

Redmine - Feature #3232

Using several environments

2009-04-23 21:59 - Leandro Freidenberg

Status:	Closed	Start date:	2009-04-23
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Administration	Estimated time:	0.00 hour
Target version:			
Resolution:	Invalid		
Description			
<p>Hello, I'm new on Redmine, Ruby, etc.</p> <p>I just need information on how to use several environments like dev, test, prd and how to migrate from one to another. I have searched everywhere in this page and I can't find related information.</p> <p>What I need is to have diferent environments to develop and test without afecting production system, and how do I migrate changes from one to another.</p> <p>I have tried setting up Development and Production environments using "ruby script/server webrick -e <env> -p <port>. I have the 2 DB created and both environments running, but I don't know how to change things for the different environmnets and to transport from one to another. Same with plugins. Want to install for one, test and then implement for the other.</p> <p>I'll appreciate all the information you can give me.</p> <p>Regards,</p>			

History

#1 - 2009-04-24 08:29 - Zarooba Rozruba

If you trust developers with passwords to repositories (I use read only accounts to repositories from redmine) :

1) setup a cron job that dumps production into dev environment

The idea behind this is to ensure that what you develop is quite close to what you see in production. Caveat: always ensure and double check that what you are about to do is on your devsite, as mistaking a dev vs prod is easy. Best: use alternate theme for dev site, so less confusion.

- mysqldump proddb > proddb.sql
- cat proddb.sql | mysql devdb

2) migrate restored schema to dev redmine settings (rake migrate?)

- rake db:migrate RAILS_ENV="dev"

this will update schema diffs between your prod (older schema, newest data), to what is required by the dev. Now oyu will have newest data + newest schema.

3) dev more

now you can do more of dev

your testing environment will be similiar, except, less cron job, and more controlled about db/file changes.

cheerios

#2 - 2009-04-24 15:52 - Leandro Freidenberg

Thanks Zarooba.

I don't really see the point of a cron job, anyway it can be usefull. Using diferent themes is usfull, and I'm already doing so.

Maybe I don't understand very much how exactly the RAKE MIGRATE works.

The point is, I don't see diferent files for each environment in directories, so don't know the moment I modify something, how to replicate that into

another environment. For example, plugins should be placed on `</redmine>/vendor/plugins/`, but how do I tell for which environment it is for so that it is developed/tested without affecting each other or even production. Same if modifying other files, code, options.

From the beginning, the idea is to have 3 different environments dev,test,prd all the same.

Over time, the idea is to develop/change/install/etc only on dev, then replicate in test, and then to prod. Is this the intention of the environments ? How do I configure/intall redmine to work this way ?

Thanks,

#3 - 2009-04-24 17:42 - Zarooba Rozruba

well, no

Redmine built in environments are to run same code with different dbs. This is good for two types of scenarios:

- perform tests on same code as prod, but with different database.
- run multiple prod environments (one code base, but different dbs). *say, you got 10 independent dev teams, and each dev team manages 10 projects*

I you want to run different codebases then you need separate redmine directories (someone please correct me). Rake migrate is designed to help you migrate the schema. My example was more to deal with upgrading Redmine. Say, you are running 0.7.3 in prod. You want to test 0.8.3. First you will backup prod. Then restore it onto your test db. Then you will run rake-migrate to upgrade db schema from 0.7.3 to 0.8.3, then you will run redmine 0.8.3 pointed to your test db.

Notice: here you have different redmine code base, each pointing to different db.

cheerios

#4 - 2009-04-24 19:32 - Leandro Freidenberg

As far as I understand, what you're saying is quite the same. If it's another version, it's similar to changing the code/installing new plugins/etc my self. New versions surely include this kind of things.

You mean that in this scenario, I should install redmine in a new directory ?

Eg: I'm running 0.7.3, need to migrate to 0.8.3, I will install 0.8.3 in another directory, copy the DB, rake-migrate to upgrade db schema.

I thought of this scenario, but thought that environments have other improved functionality. The idea was to perform tests on different code as prod, so prod didn't get affected by any changes not fully tested. And one's ready, replicate changes.

I think I should try with different installations.

Thanks

#5 - 2009-04-25 13:38 - Jean-Philippe Lang

- *Status changed from New to Closed*

- *Resolution set to Invalid*